

<<GDI+高级编程>>

图书基本信息

书名：<<GDI+高级编程>>

13位ISBN编号：9787302212942

10位ISBN编号：7302212945

出版时间：2010-1

出版时间：清华大学出版社

作者：杨建昌

页数：453

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;GDI+高级编程&gt;&gt;

## 前言

GDI+ ( Graphical Device Interface Plus ) 伴随着.NET诞生已经有好几年了。随着.NET框架的升级, GDI+的功能也越来越完善。

作为Windows的显示子系统, 它负责完成Windows窗体和控件的绘制工作, 是Windows XP以及Windows 2003等操作系统的用户界面层核心。

与它的前身GDI图形系统相比, 它引入了2D图形的反锯齿、渐变画刷、基数样条、浮点数坐标, 以及Alpha混合支持, 并支持多种图像格式等。

但是, GDI+没有任何硬件加速, 这在性能上造成了一些损失。

因此, 在某些场合仍然需要采用GDI与GDI+混合编程的模式以获得最优的性能。

在编程模式上, GDI+基于非状态的图形对象的用法使程序员可以简单地以面向对象的编程方式去完成每一个绘图任务, 而不必花费大量精力去关注各种句柄和设备上下文的状态, 这样的好处是显而易见的。

在Windows Vista操作系统中, 提供了一种全新的显示子系统——WPF。

WPF提供了更加丰富和高效的多媒体应用。

但是, GDI+作为当今Windows Forms应用程序的主流图形库仍然会在相当长的一段时间内存在。

GDI+图形库中包含了大量的类型、结构和枚举定义, 本书没有逐一介绍每一个类, 也没有介绍GDI+图形库中的每一个相关类型。

本书立足GDI+的巧妙运用, 以创建实用的、耐用的用户界面为例来介绍GDI+图形库的各个部分和功能。

本书采用C#语言描述, 不是面向.NET的初级学者的, 而是面向对.NET框架有一定认识并有一定编程经验的中高级读者。

另外, 对GDI+和C#感兴趣的在校师生也可以选择阅读本书。

1. 本书内容 本书共分为7章, 每一章以特定主题为中心、以贯穿始末的程序案例为主线讲解相关知识。

每一章中都给出了大量的源程序清单。

其中, 大部分源程序都具有极强的实用性, 可以直接应用于相关的项目中。

出于对篇幅的考虑, 在书中介绍相关的知识点时, 只会罗列核心的代码清单, 完整的源程序可以在随书附带的光盘上找到。

全部的源程序都已经在Visual Studio 2005中编译通过, 在Visual Studio 2008上通常也可以直接运行。

第1章介绍GDI+的概念, 以及层次结构, 并介绍Visual Studio 2005集成开发环境的常用工具和操作, 以及C#语言2.0版本所提供的一些新特性。

最后介绍GDI+图形库的简单结构, 并用一个小例子说明使用GDI+图形库进行绘图的简单过程。

第2章介绍GDI+的一般基础性应用——Windows基础控件的制作, 包括带有视觉样式和没有视觉样式的基础控件, 介绍分层窗口的概念和应用, 并给出利用分层窗口实现Alpha混合的异形窗口的简单封装。

本章还用了比较大的篇幅来介绍有关颜色空间处理的知识, 这可以用于增强GDI+比较弱的颜色支持。本章使用的GDI+技术包括线性渐变画刷、路径渐变画刷、路径、区域、剪辑和失效、自定义光标等。

第3章介绍更加高级的自定义创意控件的一般流程和方法。

与第2章介绍的传统的Windows基础控件相比, 本章的内容更加新颖。

利用本章介绍的知识, 就可以使用GDI+完成高级自定义控件的界面绘制和与用户交互的处理。

然后介绍利用.NET框架实现控件对RAD的支持, 并简单介绍Visual Studio IDE的开放结构与创建自定义窗体设计器的步骤。

最后, 以创意控件的制作为目标介绍控件在用户交互时对音效的支持。

本章涉及的内容较多, 例如GDI+坐标系统、矩阵的仿射变换、双倍缓存等以及相关的一些.NET技术。

第4章介绍GDI+的另一个重要方面, 即对字体、文本和打印的支持。

## &lt;&lt;GDI+高级编程&gt;&gt;

首先介绍与字体相关的一些概念，以及当今常用的字体格式，例如TrueType、OpenType，以及字体渲染技术。

然后介绍如何使用GDI+进行文本的格式化输出，包括文字修整和排版。

接下来介绍如何绘制特效文本的技术：首先是对文字进行精确的大小测量，以便对文字的绘制坐标进行精确的计算，其次是建立3D的特效文本和将文字沿路径排版的技术。

在本章的最后介绍使用GDI+进行格式化打印的技术。

第5章 介绍GDI+在处理动画上的应用。

本章将介绍一些可以有效减少使用GDI+处理动画过程中出现的闪烁现象的算法和技术，以及如何突破GDI+性能瓶颈。

具体的内容分为播放GIF动态图像、幻灯片效果实现、屏幕保护程序以及游戏。

在本章提供的源程序清单中，包含大量实用的、有趣的、新颖的算法和技术。

第6章 介绍分形和分形艺术的概念，以及分形在现实生活中的应用。

由于分形的快速发展，以及广阔的应用前景，希望通过本章的介绍能将读者引入探索奇幻莫测的分形世界的艺术殿堂。

在内容的安排上，首先简单介绍分形的概念和相关的理论，然后结合GDI+技术实现一个分形图像的生成模型，以及分形世界中最美轮美奂的复动力系统。

第7章 介绍数字图像处理的相关技术。

本章中给出了大量的专业级数字图像处理应用。

首先简单介绍GDI+提供的数字图像处理功能接口，以及其一般用法，并在此基础上结合Exif规范实现通过GDI+图形库对数码照片Exif信息的读取和写入。

接下来介绍各种数字图像处理的原理和一般算法，例如图像的点运算，包括Gamma校正、亮度与对比度调节、色相与饱和度调节、ColorMatrix的使用以及图像的直方图与直方图均衡算法等，并介绍滤波和压缩中使用较多的更加高级的图像变换原理和算法实现，以及图像的邻域运算和卷积操作。

然后系统性地介绍图像色彩混合中使用的各种混合模式，这对于多个图像的混合叠加是非常强大而有用的功能。

最后介绍图像后期处理中使用的一些特效滤镜的算法原理和实现，并探讨数字图像处理系统中相关的一些基本技术，如用户选区的平滑与羽化等。

## <<GDI+高级编程>>

### 内容概要

本书立足GDI+的巧妙运用，以制造实用的用户界面为目标来介绍GDI+图形库的各个部分和功能。

本书共分为7章，每一章介绍了特定主题内容。

全书主要内容包括GDI+的概念，以及层次结构，GDI+的一般基础性能应用——windows基础控制的制作，包括带有视觉样式和没有视觉样式的基础控件；高级自定义创意控件的一般流程和方法；GDI+对字体和文字的支持；GDI+在处理动画上的应用；分形和分形艺术的概念，以及分形在现实生活中的应用；数字图像处理的相关技术。

本书采用C#语言描述，面向对.NET框架有一定认识并有一定编程经验的中高级读者。

本书也可以作为对GDI+和C#感兴趣的在校师生的参考读物。

## 书籍目录

第1章 .NET Framework 2.0图形设备接口 1.1 GDI+Vs GDI、 DirectX、 OpenGL 1.2 Visual Studio 2005集成开发环境与Visual C # 1.3 System.Drawin9.dll程序集 1.4 本章小结第2章 Windows视觉样式和WinForm基础控件 2.1 VisualStyles和ControlPaint类 2.1.1 视觉样式浏览器 2.1.2 ControlPaint类 2.1\_3 Windows导航栏控件 2.2 分层窗口 2.2.1 分层窗口的概念 2.2.2 CreateParams类 2.2.3 使用分层窗口 2.3 颜色空间和拾色器 2.3.1 颜色空间 2.3.2 色彩转换 2.3.3 拾色器 2.4 本章小结第3章 GDI+与.NET创意控件 3.1 不规则形状 3.1.1 Circle类 3.1.2 Circle类型转换器 3.1.3 高级衍生图形 3.1.4 WaitingCircle控件 3.2 坐标系统与度量单位 3.2.1 坐标系统 3.2.2 屏幕坐标与工作区坐标 3.2.3 度量单位 3.2.4 Ruler控件 3.3 矩阵变换 3.3.1 Matrix类 3.3.2 矩阵的仿射变换 3.3.3 全局变换和局部变换.....第4章 字体、文本和打印第5章 动画第6章 分形艺术 第7章 图像处理附录

## 章节摘录

OpenGL ( Open Graphics Library ) 是由SGI ( Silicon Graphics Inc. ) 公司开发的一套开放图形库。它是一套跨平台、不依赖于编程语言的2D / 3D图像编程接口, 应用在包括CAD、GIS、媒体娱乐、游戏开发、工程科学及虚拟现实等行业领域中。

与DirectX相比, OpenGL只是一个图形函数库, 并不包含音视频处理、输入设备管理、网络连接等支持API。

OpenGL以强大的功能、良好的移植性、高效的图像处理方法以及硬件扩展等特性在图形工作站等专业高端图形处理领域得到广泛应用而成为事实上的行业标准。

由于Microsoft为了使DirectX在游戏开发市场竞争中更具有优势, 于是采取了一系列措施来限制OpenGL的发展, 例如拒绝在操作系统中发布支持OpenGL的驱动程序等。

OpenGL的发展一直处于一种近乎停滞的状况, 而DirectX的发展则是蒸蒸日上, 在PC领域成了完全的霸主。

虽然如此, OpenGL在高端图形处理领域仍旧是豪气十足, 不可取代。

DirectX与OpenGL虽然功能强大, 但是其复杂程度是GDI+远远不能比拟的。

DirectX与OpenGL属于重量级的图形库, 而GDI+只是轻量级的图形库, 但是却提供了友好简单的编程界面, 易学易用。

DirectX与OpenGL面向三维图像以及音视频等多媒体处理, 直接面向显卡编程, 比较适合大型多媒体应用程序, 比如游戏、三维建模工具等的开发, 在二维图像处理方面没有提供绘图接口和函数, 只有位图操作。

GDI+则集成了众多图形相关的类, 并且提供了一组相当丰富的绘图函数, 在Windows窗口以及桌面编程领域, 可以完全胜任二维图元的绘制操作, 也是Windows操作系统中窗口编程的绝对主力。

DirectX、OpenGL以及GDI+等图形库都有各自最适合的应用领域。

由于GDI+更注重系统的简洁性和兼容性, 从而更易学易用。

如果应用程序或游戏中不要求太多的3D效果, GDI+同样可以非常适合该类应用程序和很多类型游戏的开发。

在常规的Windows窗口应用程序中, 使用GDI+是最好的选择。

结合程序员对GDI+图形库的扩展, GDI+可以在多个应用领域进行多种图形图像、地理信息、包装装潢、广告设计, 以及游戏等应用的开发。

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>